

REMARKS

The Office Action dated August 8, 2005, has been received and carefully noted. The above amendments to the claims, and the following remarks, are submitted as a full and complete response thereto.

Claims 1 and 7 are amended to more particularly point out and distinctly claim the subject matter of the invention. Support for the changes to claims 1 and 7 may be found in the specification, for example, on page 100, lines 7-21 and page 103, lines 20 and 21. No new matter is added by these amendments. Thus, claims 1-7 are pending in the present application, and are respectfully submitted for consideration.

Claims 1, 2 and 7 were rejected under 35 U.S.C. § 102(a) as being anticipated by *Muller et al.* (U.S. Patent No. 5,909,686 or *Muller '686*). Claims 3-6 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Muller '686* in view of *Muller et al.* (U.S. Patent No. 6,119,196 or *Muller '196*). The above rejections are respectfully traversed according to the remarks that follow.

Claim 1, upon which claim 2 is dependent, recites a network switch stack configuration. The configuration includes a first network switch comprising a plurality of data ports, a first stacking port, a first internet port interface controller, and a first CPU interface. The configuration also includes a set network switch having a plurality of data ports, a second stacking port, a second internet port interface controller, and a second CPU interface. The configuration also includes a common CPU connected to the first CPU interface and the second CPU interface. The first stacking port and the second

stacking port are communicatively connected through the first and second internet port interface controllers, such that incoming packets on any of the plurality of data ports on the first and second switches are effectively switched to any of the plurality of data ports on either of the first and second network switches. The first switch adds a module header having module identifier fields, providing a source module ID of the first switch, to the incoming packets and the second stacking port reads the module headers to determine egress ports for the packets. Claims 2-6 are dependent upon claim 1.

Claim 7 recites a method for routing packets in a network switch stack configuration. The method includes communicatively connecting a first stacking port and a second stacking port. A first network switch includes the first stacking port and a second network switch includes the second stacking port. The method also includes adding module headers having module identifier fields, providing a source module ID of the first switch, to incoming packets on any of a plurality of data ports by said first switch and reading the module headers by the second stacking ports to determine egress ports for said incoming packets. The method also includes switching the incoming packets to the egress ports via at least one internet port interface controller. The egress ports include any of the plurality of data ports on either of the first and second network switches.

Muller '686 is directed to a method and apparatus for providing hardware-assisted CPU access to a forwarding database. A switch fabric provides access to a forwarding database on behalf of a processor, and includes a memory access interface configured to arbitrate access to a forwarding database memory. The switch fabric includes interfaces

for communicating with a CPU, shared memory, network ports and a cascading interface communicating with one or more switch elements.

Muller '196 is directed to a method and apparatus for managing a buffer memory in a packet switch that is shared between multiple ports in a network system. The apparatus comprises a plurality of slow data port interfaces configured to transmit data at a first data rate between a slow data port and the buffer memory and a plurality of fast data port interfaces configured to transmit data at a second data rate between a fast data port and the buffer memory. A first level arbiter is coupled to the plurality of slow data port interfaces, where the first level arbiter chooses an access request of one the slow data ports and outputs the access request.

Applicants note that the discussion of the module header and its fields makes clear that at least one purpose of the module header is to identify the source module that the packet is received on. As discussed on page 103, lines 20 and 21, the module header contains a field that provides a module ID for the source switch which received the packet. This should be distinguished from a generic header, which it is acknowledged that most packets would have when they are forwarded on a network. Thus, claim 1 recites, in part, “wherein the first switch adds a module header having module identifier fields, providing a source module ID of the first switch, to the incoming packets and the second stacking port reads the module header to determine egress ports for the packets,” where a similar limitation can be found in claim 7. Applicants respectfully assert that the above-cited prior art references fail to teach or suggest such a limitation.

The instant specification clearly describes the function and makeup of a module header, including module identifier fields, according to one embodiment of the present invention. Page 100, lines 29-32, discloses that “[t]he module header information, appended to the packet by the source, is received by IPIC 90 from the P channel. The module header includes the module ID bitmap, COS, mirrored-to port/switch information, trunk group ID, etc. The constructed packet is then sent onto the high performance interface 261.” A full description of composition of the module header is provided in the specification of the instant application at page 102, line 4 to page 103, line 29. Thus, Applicants respectfully assert that the recitation of module header, including module identifier fields, in claims 1 and 7 does not read on a generic header and that *Muller* ‘686 fails to teach or suggest such an element. For at least this reason, Applicants respectfully assert that the anticipation rejection of claims 1 and 7 is improper and should be withdrawn.

Additionally, Applicants also respectfully traverse the Office’s position that a header is inherently added to each incoming packet. While it could conceivably be asserted that *Muller* ‘686 discloses the use of fields of the packet to determine an output port, *Muller* ‘686 fails to teach or suggest adding a header to an incoming packet. While the rejection appears to allege that such a function would be “inherent,” Applicants note that the use of inherent teachings in rejections is not so liberal. “Inherency, however may not be established by probabilities or possibilities. The mere fact that a certain thing *may* result from a given set of circumstances is not sufficient.” Continental Can Co. USA Inc.

v. Monsanto Co., 20 USPQ2d 1746, 1749 (Fed. Cir. 1991). As such, for this additional reason, Applicants respectfully assert that the anticipation rejection of claim 1 is improper and should be withdrawn.

Thus, Applicants respectfully assert that any rejection of claims 1 or 7 over *Muller* '686 and/or *Muller* '196 would be improper for failing to teach or suggest all of the elements of that claim. On the basis of the above, independent claim 1 is respectfully asserted to be patentable, and as a consequence the dependent claims 2-6 are patentable as well. It is therefore respectfully requested that claims 1-6 be allowed and this application be allowed to pass to issue.

In the event this paper is not being timely filed, the applicants respectfully petition for an appropriate extension of time. Any fees for such an extension together with any additional fees may be charged to Counsel's Deposit Account 50-2222.

Respectfully submitted,



Kevin F. Turner
Registration No. 43,437

Customer No. 32294
SQUIRE, SANDERS & DEMPSEY LLP
14TH Floor
8000 Towers Crescent Drive
Tysons Corner, Virginia 22182-2700
Telephone: 703-720-7800
Fax: 703-720-7802

KFT:jf